A GEOFENCE VIOLATION PREVENTION MECHANISM FOR SMALL UAS

Swee Balachandran* , **Anthony Narkawicz** , **César Muñoz** , **María Consiglio****
***National Institute of Aerospace, Hampton, Virginia 23666** ,
****NASA Langley Research Center, Hampton, Virginia 23666**

Keywords: *Small UAS, Geofencing, Referemce Governor, Formal Verification*

Abstract

The ability to safely confine the trajectories of small UAS to a specific geographical area is a key enabler for missions that require operating in close proximity to populated areas as well as other users of the airspace. These capabilities require highly reliable geofencing algorithms. In particular, these algorithms must promptly alert imminent breaches of keep-in/keep-out geofences by considering factors such as the vehicle speed and uncertainties in the state of the aircraft. This paper presents a novel approach to the prevention of geofence boundary violation based on closure rate constraints. These constraints are incorporated into a control framework to effectively prevent fence breaches. Simulation results illustrating an example use case of this framework are presented.

1 Introduction

The increased usage of small unmanned aircraft systems (UAS) by hobbyists and commercial operators has resulted in a growing number of safety and regulatory concerns. Operational anomalies such as loss of remote control (RC) or telemetry link with the UAS can result in the operator losing the ability to control the vehicle. This could lead to the vehicle flying away and posing risk to life forms and property on the ground, and also other users of the airspace.

Geospatial constraints called geofences help impose virtual boundaries that restrict the area of operation of small UAS vehicles. Geofences help uphold the safety of UAS operations, protect civilian and other life forms, prevent destruction to property. Geofencing has grown to become an essential functionality for small UAS. Indeed, most commercial-off-the-shelf autopilot systems today incorporate some kind of geofencing functionality. These geofencing systems provide the operator with warnings about fence breaches and take appropriate actions such as return to home or land to prevent fly aways. Furthermore, independent geofencing modules that can be added on to existing autopilots are currently under development.

The growing regulatory emphasis on geofencing systems and their associated safety implications make it desirable to have assurances of their correct behavior. It is also important to understand the effects of discrete sensor update rates and uncertainties on the behavior of geofencing systems. In particular, these can affect the selection of appropriate mission parameters. This paper uses a modeling framework that incorporates discrete sensor updates and position uncertainties. Several of the formulas presented in this paper are formally proved in the Prototype Verification System (PVS) [1]. The use of formal verification tools, such as PVS, provides high assurance on the correctness of the mathematical development presented in this paper.

This paper provides an algorithm for geofence violation detection and prevention. Specifically, this paper studies the relevant constraints that must be imposed on a vehicle's

velocities to prevent a fence breach, taking into account the dynamics of the vehicle, performance, and sensor uncertainties. This analysis naturally lends itself to the construction of a command filtering scheme, typically referred to as a reference governor, that can be added on to a vehicle's control law to effectively prevent geofence violations. This paper is organized as follows: Section 2 surveys related work. Section 3 constructs the required constraints that must be considered to prevent fence breaches. Section 4 uses the proposed constraints to construct a reference governor filter scheme that helps resolve any imminent fence breaches. Section 5 presents simulation results on a use case. Section 6 provides discussions and Section 7 concludes this work.

2 Related Work

The notion of geofencing seems to first appear in the context of location based services and notifications [2]. Unlike location based services, the widespread adoption of geofencing technology in safety-critical applications such as the operation of small UAS, warrants careful analysis of the properties of these algorithms to ensure that they perform as expected for all possible input configurations and initial conditions.

The use of geofences as an effective safety tool to enforce airspace constraints was emphasized in [3]. Recognizing the safety-critical role played by geofences for small UAS, the author outlines an architecture that envisions a geofencing system independent from the primary autopilot. This system would run on a secondary processor thus providing redundancy if the autopilot were to malfunction. Similar ideas were also identified for assured airspace containment in [4].

Narkawicz et al. rigorously analyzed the properties of ray-casting based algorithms to determine if a given point in 3D space is located within a polygon [5, 6]. The algorithms were modeled in PVS, and several important properties were formally proved in that system. An implementation of these algorithms called PolyCARP is provided in several programming languages. Dill et al. proposed an independent geofencing system called Safeguard [7–9], which uses an alternative implementation of the PolyCARP algorithms. Safeguard, which consists of a self-contained hardware and sensor suite, serves as an external geofence system constantly monitoring for imminent violations using a ballistic trajectory model. The outputs of Safeguard can be configured to terminate flight or implement other appropriate responses.

In line with the architecture discussed in [3], Stevens et al. developed an implementation of a platform independent add-on geofencing system for small UAS and demonstrated its feasibility on an off-the-shelf autopilot system [10]. The independent goefencing system had three different user configurable modes of operation [11]: (i) shared control, (ii) return to land, and (iii) local loiter. In [12], the authors formally define the notion of geofencing and outline the data requirements of a geofence. Furthermore, in [13], the authors evaluate the various geofencing algorithms such as ray casting and triangle weight approximations, and explore their computational complexities.

The work presented in this paper proposes two fundamental developments that aid the construction of efficient geofencing systems for rotorcraft UAS. First, constraints that must be enforced on the aircraft's velocities to prevent geofence violations are analyzed as a function of the vehicle's position and heading within the geofence. In the second development, the constraints developed above are utilized to generate a filtering mechanism that eliminates control inputs that could lead to geofence violations. The second development is in line with the shared control scheme proposed in [11]. The detailed framework presented in this work applies to general geofences rather than just rectangular shapes.

3 Geofence conflict detection

There are two primary components to an onboard geofencing system: (1) conflict detection and (2) resolution. Conflict detection for geofencing is concerned with monitoring the vehicle's posi-

tion for imminent fence breaches taking into account the dynamics of the vehicles, sensor uncertainty and and mission constraints. Resolution is concerned with reacting to an imminent fence breach to mitigate any risk involved and prevent the fence breach.

There are several steps required to determine if an object moving at a certain speed can result in a fence breach. The capability to recognize if a given point is within the fence (a polyhedron) is prerequisite for conflict detection. The problem of finding if a given point in 3-D space is located inside or outside a polyhedral volume of airspace was considered in [5, 6]. One method used in those algorithms is ray casting. Given a 2D point, a ray is cast from the given point to infinity. If the line makes even (odd) number of crossings with the polygon edges, the point is outside (inside) the polygon. Formal properties of these algorithms were verified using PVS [6].

3.1 1-D Analysis of constraint violation

Consider a particle moving in 1D towards a point P. Let d represent its proximity to P and c represents its closure rate to P. The position of the particle is sampled at regular intervals Δ_p. Let σ represent the uncertainty (standard deviation) in the particle's position. The estimated distance moved by the particle between position updates is given by

$$D_{gps} = c\Delta_p + 2\sigma. \tag{1}$$

Assuming the goal is to stop the particle before crossing P by applying a deceleration of $\bar{a}$, the maximum distance moved by the particle before stopping is given by

$$D_{dec} = \frac{c^2}{2\bar{a}}. \tag{2}$$

A timely warning generated to prevent the particle from crossing P should take into account the distance moved by the particle between position updates and the distance moved during the response applied to prevent violation. Consequently, a monitoring algorithm to prevent the particle from crossing P should at least provide

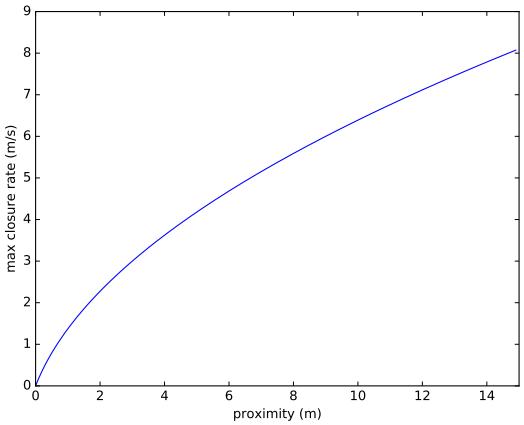

Fig. 1 Proximity to edge vs maximum closure rate

a warning when the particle is at a distance of d^* from the point P, where

$$\begin{aligned} d^* &= D_{gps} + D_{dec}, \\ d^* &= c\Delta_p + 2\sigma + \frac{c^2}{2\bar{a}}. \end{aligned} \tag{3}$$

Alternately, the maximum allowable closure rate at any given distance d from point P is given by the positive root of the following quadratic equation:

$$\frac{c^2}{2\bar{a}} + c\Delta_p + 2\sigma - d = 0. \tag{4}$$

Figure 1 illustrates the relation between the proximity distance d and the maximum closure rate $c^*(\bar{a}_{max}, d)$ computed using equation (4). Thus by enforcing the constraint

$$c \leq c^*(\bar{a}, d) \tag{5}$$

a timely warning can be generated to prevent the particle from crossing point P.

3.2 2-D analysis of constraint violation

Let $\vec{x} = [x, y]$ represent the particle's position and let $\vec{v} = [v_x, v_y]$ represent the particle's velocity in 2D space. Let PQ represent a line segment, and let `r` (short-hand for *restrict*) be a Boolean indicating whether the algorithm should ensure separation from the infinite line containing PQ (in which case `r = false`) or from the restricted

line segment PQ (in which case $\mathtt{r} = \mathtt{true}$). Let $d_{\mathtt{r}}$ represent the distance of the particle to this line (when $\mathtt{r} = \mathtt{false}$) or line segment (when $\mathtt{r} = \mathtt{true}$). Let $\hat{n}_{\mathtt{r}}$ represent the unit vector from $\vec{x}$ to the closest point on this line or line segment. For example, if $\vec{x} = [-2,1]$ and PQ is the line segment between the points $[-1,0]$ and $[1,0]$, then $\hat{n}_{\mathtt{false}} = [0,-1]$, and $\hat{n}_{\mathtt{true}} = [1/\sqrt{2},-1/\sqrt{2}]$. When PQ is a segment on the boundary of the restricted region, $\hat{n}_{\mathtt{r}}$ points in the direction of the region. The closure rate ($c_{\mathtt{r}}$) of the particle to line or line segment is given by

$$c_{\mathtt{r}} = \hat{n}_{\mathtt{r}} \cdot \vec{v}. \tag{6}$$

Thus, enforcing the constraint (5), where the closure rate c is determined by (6), results in the constraint

$$c_{\mathtt{r}} \leq c^*(\hat{n}_{\mathtt{r}} \cdot \vec{a}, d_{\mathtt{r}}), \tag{7}$$

where $\vec{a}$ is the deceleration *vector* of the particle. It has been formally proved in PVS that in the constraint (7) is sufficient to guarantee separation from the line segment P, at least in the case of negligible update rates and position uncertainty.

3.3 Application to polyhedron

Consider a keep-in polygon with N edges. As above, let $\hat{n}_{ri}$ represent the unit vector in the direction of edge $i \in \{1 \ldots N\}$, and let $d_{\mathtt{r}i}$ denote the distance to the line containing edge i (when $\mathtt{r} = \mathtt{false}$) and edge i (when $\mathtt{r} = \mathtt{true}$). A timely warning to prevent a particle from exiting the convex keep-in polygon can be generated by enforcing the constraint 7 on all edges. Thus,

$$c_{\mathtt{r}i} = \hat{n}_{\mathtt{r}i} \cdot \vec{v}, \ \forall i \in \{i = 1, \ldots, N\}, \tag{8}$$
$$c_{\mathtt{r}i} \leq c^*(\hat{n}_{\mathtt{r}} \cdot \vec{a}, d_{\mathtt{r}i}). \tag{9}$$

For convex keep-in regions, constraint (9) is used with $\mathtt{r} = \mathtt{false}$. For non-convex polygons and keep-out regions, constraint (9) is used with $\mathtt{r} = \mathtt{true}$. However, for these polygons, it is often difficult or impossible to enforce constraints (9) for each edge simultaneously. Alternately, constraint (9) is enforced only on the active edges for which $\hat{n}_{\mathtt{r}i} \cdot \vec{v} \geq 0$. Here, an edge i is considered active if $d_{\mathtt{false}i} = d_{\mathtt{true}i}$. However, note that if the nearest point on the nearest edge corresponds to a vertex (common point between two adjacent edges), then it is sufficient to check (9) where the closure rate $c_{\mathtt{r}i}$ (given by (6)) is defined with respect to the averaged normals of the adjacent edges.

4 Geofence Conflict Resolution

These constraints discussed above can be used to construct a command filtering mechanism to eliminate control inputs that can lead to fence breaches. The following sections first discuss the preliminaries of a command filtering mechanism called a reference governor [14, 15]. Subsequently, the reference governor scheme is applied to prevent fence breaches.

4.1 Reference Governors

Consider a closed-loop discrete-time linear system defined by

$$\mathtt{x}(t+1) = A\mathtt{x}(t) + B\mathtt{v}(t), \tag{10}$$
$$\mathtt{y}(t) = C\mathtt{x}(t). \tag{11}$$

where $\mathtt{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathtt{v}(t) \in \mathbb{R}^m$ is the filtered input vector, and $\mathtt{y}(t) \in \mathbb{R}^p$ is the output vector. Here, the closed-loop system is assumed asymptotically stable, (i.e., all eigen values of A are strictly in the interior of the unit disk). Constraints are imposed on the output variables $\mathtt{y}(t)$.

At each time instant, reference governors compute a command $\mathtt{v}(t)$ such that, if kept constant from t onwards, the resulting output will always satisfy the constraints[1]. Let O_∞ define the set of all states $\mathtt{x}$ and inputs $\mathtt{v}$ such that the predicted response from the initial state $\mathtt{x}$ with a constant input $\mathtt{v}$ satisfies constraints.

$$O_\infty = \{(\mathtt{v},\mathtt{x}) : \hat{y}(k|\mathtt{v},\mathtt{x}) \in Y, \forall k \in \mathbb{Z}\}, \tag{12}$$

[1]For the dynamical system in this work where one or more eigen values are on the unit disk, constraint satisfaction is imposed up to a time horizon.

where the predicted response is given by

$$\hat{\mathrm{y}}(k|\mathrm{v},\mathrm{x}) = CA^k\mathrm{x} + C\sum_{j=1}^{k} A^{j-1}B\mathrm{v} \tag{13}$$

$$= CA^k\mathrm{x} + C(I-A)^{-1}(I-A^k)B\mathrm{v}. \tag{14}$$

Here Y represents the constraint admissible set for the outputs.

Given the actual input $\mathrm{r}(t)$, let the filtered $\mathrm{v}(t)$ be parametrized as

$$\mathrm{v}(t) = \mathrm{v}(t-1) + \beta(t)(\mathrm{r}(t) - \mathrm{v}(t-1)). \tag{15}$$

At each discrete time step, the following optimization problem:

$$\beta(t) = \max_{\beta\in[0,1]} \beta \tag{16}$$
$$s.t.\ \ (\mathrm{v},\mathrm{x}(t)) \in O_\infty.$$

is solved to compute the filtered input commands that can prevent constraint violation.

4.2 Application of reference governors to geofencing

Let the dynamics of the closed-loop system of a small UAS (rotorcraft) be represented by equation (10). Here, $\mathrm{x} = [x, v_x, y, v_y]$ represents the state of the vehicle. A, B represent the closed-loop system dynamics with a control law that can track a given velocity input. For example, consider the following dynamic model:

$$A = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 0.94 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 0.94 \end{bmatrix}, \tag{17}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0.1 & 0 \\ 0 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

The closed-loop system (A) is not Shur and therefore equation (14) cannot be used for predicting the output. Instead, equation (13) is used to compute the predicted output for a given horizon k. Note that the required output here is the component of the velocity perpendicular to the nearest edge, i.e., in equation (11):

$$C = [0, n_x, 0, n_y]. \tag{18}$$

Note that here n_x, n_y represent the unit normal to the nearest edge. The input v is parametrized as follows:

$$v(t) = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix}, \tag{19}$$

$$\beta \in [0,1]. \tag{20}$$

Here, r_1, r_2 represent the unfiltered velocity commands.

Given the non-linear constraints (9), β is computed using the bisection approach discussed in [15]. Note that here, constraint satisfaction is checked only up to a given prediction horizon k.

5 Results

The approach discussed above provides an efficient way of integrating geofence conflict detection and resolution into an existing control framework. Consequently, this framework is applicable to several real world use cases. As a concrete example, this work illustrates the application of the proposed framework to a structural inspection type mission.

Consider an inspection mission where an operator is manually flying a rotorcraft to acquire imagery of a structure such as a building or bridge. In such missions, it is crucial to maintain a safe stand-off distance from the structure to prevent colliding into the structure. Maintaining precise control in close proximity to the structure can be challenging. Adequate depth information can be hard to perceive from the vantage point of a remote pilot or from an onboard camera. With the proposed approach, a remote pilot can maneuver the vehicle near the structure focusing on acquiring appropriate imagery rather than being concerned about colliding into the structure.

Figure 2 illustrates a simple keep-out geofence (in red) surrounding the structure under inspection. A simple control law distorted with noise to simulate human inputs is constructed to

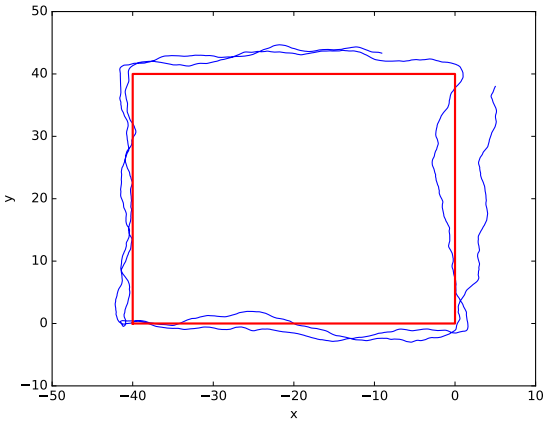

Fig. 2 Simulated pilot inputs with noise drives the vehicle around the keep-out fence (in red) representing an inspection mission. Trajectory (in blue) starts at (5,38) and moves clockwise around the fence. Intermittent fence breaches are observed at several locations.

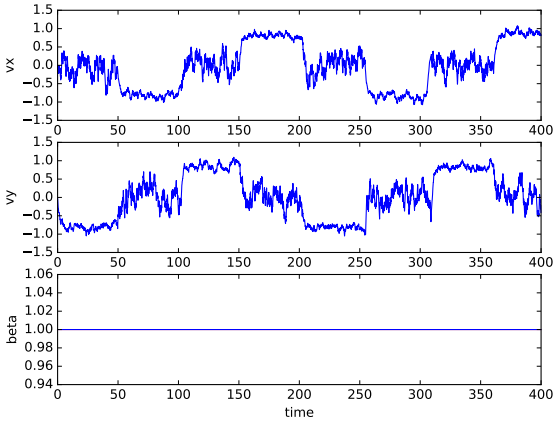

Fig. 3 Velocity (m/s) vs time (s) response of the vehicle. $\beta = 1$ implies that the original commands are passed through without any filtering.

maneuver the vehicle around the geofence thus representing a typical inspection mission. The trajectory of the vehicle is indicated in blue. Notice in Figure 2, the vehicle enters the keep-out geofence at several locations. Figure 3 illustrates the velocity components of the vehicle as it circles the geofence. Figure 4 represents the same scenario but with the simulated human inputs filtered using the scheme discussed above. Consequently, the filtering mechanism eliminates all components of velocities that can lead to the fence violation. The attenuation of the input command is represented by the β parameter indicated in Figure 5. Figures 6-9 indicate the same results on a non-convex keep-out geofence.

6 Discussion

The analysis of constraint violation in Section 3 was focused on rotorcraft vehicles. The maximum closure rate calculation assumed decelerating to a complete stop as a feasible maneuver. For fixed wing vehicles, turning away from the fence boundary using the maximum turn radius is a feasible maneuver. Consequently, equation (2) should be replaced with the turn radius used for the resolution.

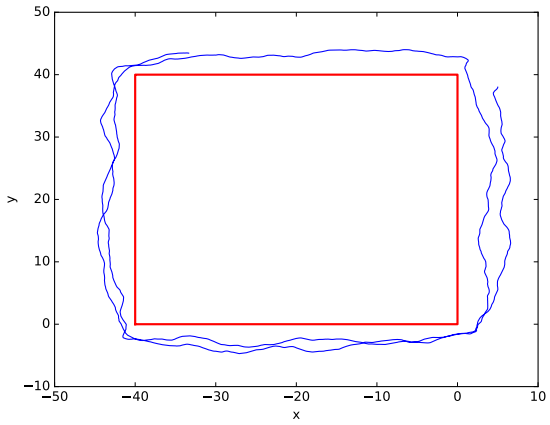

Fig. 4 Vehicle response with the reference governor scheme. Trajectory (in blue) starts at (5,38) and moves clockwise around the fence. No fence breaches are observed.

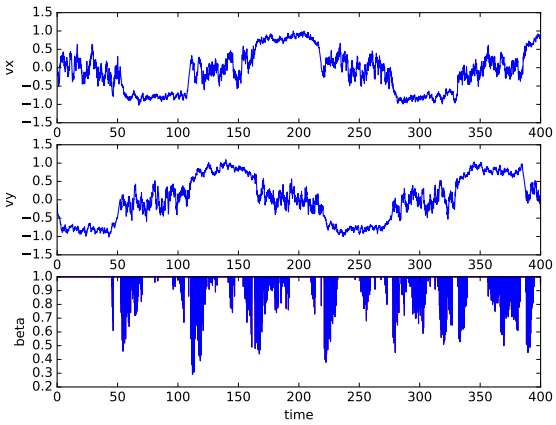

Fig. 5 Velocity (m/s) vs time (s) response of the vehicle. β indicates the amount of command attenuation.

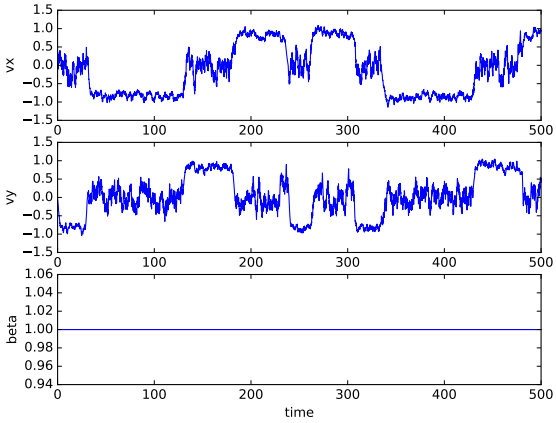

Fig. 7 Velocity (m/s) vs time (s) response of the vehicle. $\beta = 1$ implies that the original commands are passed through without any filtering.

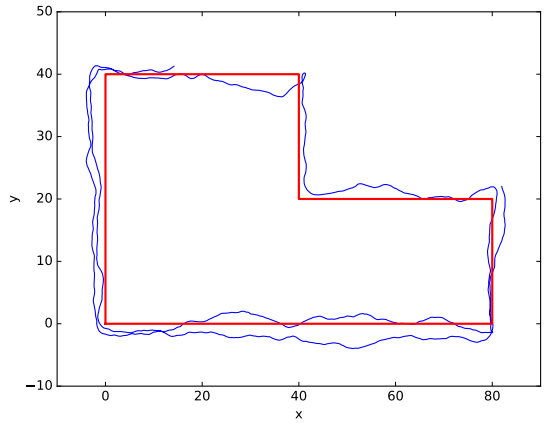

Fig. 6 Simulated pilot inputs with noise drives the vehicle around the keep-out fence (in red) representing an inspection mission. Trajectory (in blue) starts at (85,25) and moves clockwise around the fence. Intermittent fence breaches are observed at several locations.

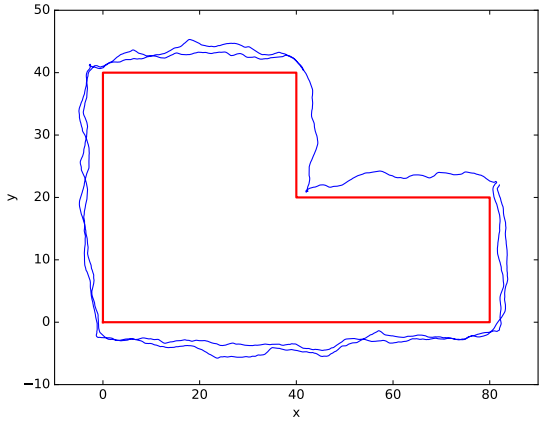

Fig. 8 Vehicle response with the reference governor scheme. Trajectory (in blue) starts at (85,25) and moves clockwise around the fence. No fence breaches are observed.

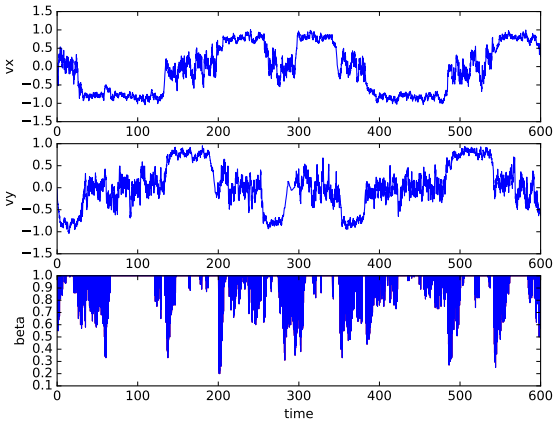

Fig. 9 Velocity (m/s) vs time (s) response of the vehicle. β indicates the amount of command attenuation.

Constructing the reference governor requires identifying the dynamic model of the closed-loop system A, B and its associated uncertainties. The formulation presented in this work did not account for any process or sensor uncertainty. However, as shown in [16], the reference governor can be modified to deal with uncertainties.

The output prediction (equation 13) requires the user to specify a look-ahead horizon. A suitable look-ahead horizon can be selected based on the application and available computing capability. The larger the look-ahead horizon, the computation of β at each time step becomes more expensive.

7 Conclusions

This work analyzed the constraints required by an effective geofence monitoring algorithm to generate timely warnings to ensure a resolution can prevent fence breaches. Constraints are imposed on the closure rate of the vehicle to a fence edge. These constraints are incorporated into a command filtering mechanism to eliminate components of the command that lead to constraint violations. Use case results illustrating the effectiveness of the proposed framework are presented.

The novel contributions of this work are the development of constraints on the closure rate of the vehicle with respect to a geofence edge, the rigorous mathematical analysis performed in the formal verification system PVS, and the utilization of the closure rate constraints in a reference governor formulation to prevent geofence violation.

References

[1] S. Owre, J. M. Rushby, and N. Shankar, "Pvs: A prototype verification system," in *International Conference on Automated Deduction*. Springer, 1992, pp. 748–752.

[2] J. P. Munson and V. K. Gupta, "Location-based notification as a general-purpose service," in *Proceedings of the 2Nd International Workshop on Mobile Commerce*, ser. WMC '02. New York, NY, USA: ACM, 2002, pp. 40–44. [Online]. Available: http://doi.acm.org/10.1145/570705.570713

[3] E. M. Atkins, "Autonomy as an enabler of economically-viable, beyond-line-of-sight, low-altitude uas applications with acceptable risk," in *AUVSI unmanned Systems*, 2014.

[4] K. J. Hayhurst, J. M. Maddalon, N. A. Neogi, and H. A. Verstynen, "A case study for assured containment," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 260–269.

[5] A. Narkawicz and G. E. Hagen, "Algorithms for collision detection between a point and a moving polygon, with applications to aircraft weather avoidance," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3598.

[6] A. Narkawicz, C. Muñoz, and A. Dutle, "The MINERVA software development process," in *Automated Formal Methods*, ser. Kalpa Publications in Computing, N. Shankar and B. Dutertre, Eds., vol. 5. EasyChair, 2018, pp. 93–108. [Online]. Available: https://easychair.org/publications/paper/g1Rs

[7] E. T. Dill, S. D. Young, and K. J. Hayhurst, "Safeguard: An assured safety net technology for uas," in *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*. IEEE, 2016, pp. 1–10.

[8] R. V. Gilabert, E. T. Dill, K. J. Hayhurst, and S. D. Young, "Safeguard: Progress and test re-

sults for a reliable independent on-board safety net for uas," in *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th*. IEEE, 2017, pp. 1–9.

[9] E. T. Dill, K. J. Hayhurst, S. D. Young, and A. J. Narkawicz, "Uas hazard mitigation through assured compliance with conformance criteria," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 1218.

[10] M. N. Stevens, B. Coloe, and E. M. Atkins, "Platform-independent geofencing for low altitude uas operations," in *15th AIAA Aviation Technology, Integration, and Operations Conference*, 2015, p. 3329.

[11] M. N. Stevens and E. M. Atkins, "Multi-mode guidance for an independent multicopter geofencing system," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3150.

[12] ——, "Geofencing in immediate reaches airspace for unmanned aircraft system traffic management," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 2140.

[13] M. N. Stevens, H. Rastgoftar, and E. M. Atkins, "Specification and evaluation of geofence boundary violation detection algorithms," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1588–1596.

[14] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[15] U. Kalabić, I. Kolmanovsky, and E. Gilbert, "Reference governors for linear systems with nonlinear constraints," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 2680–2686.

[16] E. G. Gilbert and I. V. Kolmanovsky, "Fast reference governors for systems with state and control constraints and disturbance inputs," *Int. J. Robust Nonlinear Control*, vol. 9(15), pp. 1117–1141, 1999.